MetricViews | Issue 1 | July 2021

MetricVIEWS

IFPUG.org

A PUBLICATION OF THE INTERNATIONAL FUNCTION POINT USERS GROUP

How to Spread Function Point's Word

How Can Communication and the Systemic Approach Help Us for Software Measures?

FUNCTIONAL SIZING

OF AGILE PROGRAMS AT U.S. DEPARTMENT OF HOMELAND SECURITY

PRACTICING AGILE:
SCRUM, KANBAN OR SCALING? HOW DO YOU KNOW?

By: Joe Schofield

"Please don't tell me you're doing Agile" began the article *Keep the Baby*[1] some seven years ago. That article exposed the challenge to verify claims of agility, due to variation in definition, understanding and practice. While the Agile community has evolved since the earlier days of "if it's written down, it's not Agile," the ambiguity associated with Agile terminology has continued to proliferate. As examples:

- Are we doing iterations or sprints?
- Do we conduct Sprint 0 events?
- Is a spike an increase in effort or something else altogether?
- Wait, a scrum master is no master?
- What does a minimal viable product mean?
- Is that a scrum board, a task board or a Kanban board?

Phrases like "we do Kanban" or "we are scaling" or "we use a tool" could be interpreted with a wide variation among actual practices. Would some standardization reduce ambiguities in the Agile world today? For function point advocates, do functional and non-functional measurement standards inspire more consistency or more resistance in tribal-knowledge-driven cultures?

Let's unravel some of those topics. The notion of "if it's written down, it's not Agile" is likely rooted in the second value statement of the *Manifesto for Agile Software Development*[2] which exhorts "working software over comprehensive documentation." This phrase from 2001, seems to have been an attempt to

differentiate "Agile" from more rigorous software development approaches captured in IEEE[3] or ISO[4] standards for software engineering, or then, the Software Capability Model® (SW-CMM)[5]—the forerunner to the CMMI℠.

The *Scrum Guide*[6] might serve as an example of efforts to both write something down while minimizing documentation. The sixth version of the *Scrum Guide* was released in November of 2020; however, the first version wasn't released until 2010, nine years after the development of the manifesto, and 15 years after Schwaber and Sutherland's Scrum-themed presentation at OOPSLA in Austin, Texas.[7] Compare this 15-year gap with the first Agile principle touting "early and continuous delivery."[8] While the current version of the *Scrum Guide* contains no glossary of terms, the document is itself depicted as the *definitive* guide to scrum. The *Scrum Body of Knowledge (SBoK) Guide*™[9] by contrast, has more than 50 pages of terms in its glossary. Its first release was circa 2013. An industry standard for Scrum would promote consistency of practice in the growing number of industries where Scrum is being adopted. Many of the Scrum practices in use today have evolved ancestrally from the likes of eXtreme Programming, and it from test-driven development, while others from folklore (estimation techniques using dogs, come quickly to mind). The opportunity to consolidate and perform consistently, more predictably, persists. Technology itself seems afflicted with a preference toward innovation and trendiness over leveraging and improving. No one should anticipate any attempts to standardize Agile or Scrum soon as the branches of application and the growing number of special interest groups aren't likely to converge any more than during the codification of the manifesto. The limited output of two days of work in Snowbird, Utah, that resulted in the manifesto, could be characterized as "not how much they knew about software development, but rather, how little they agreed!"

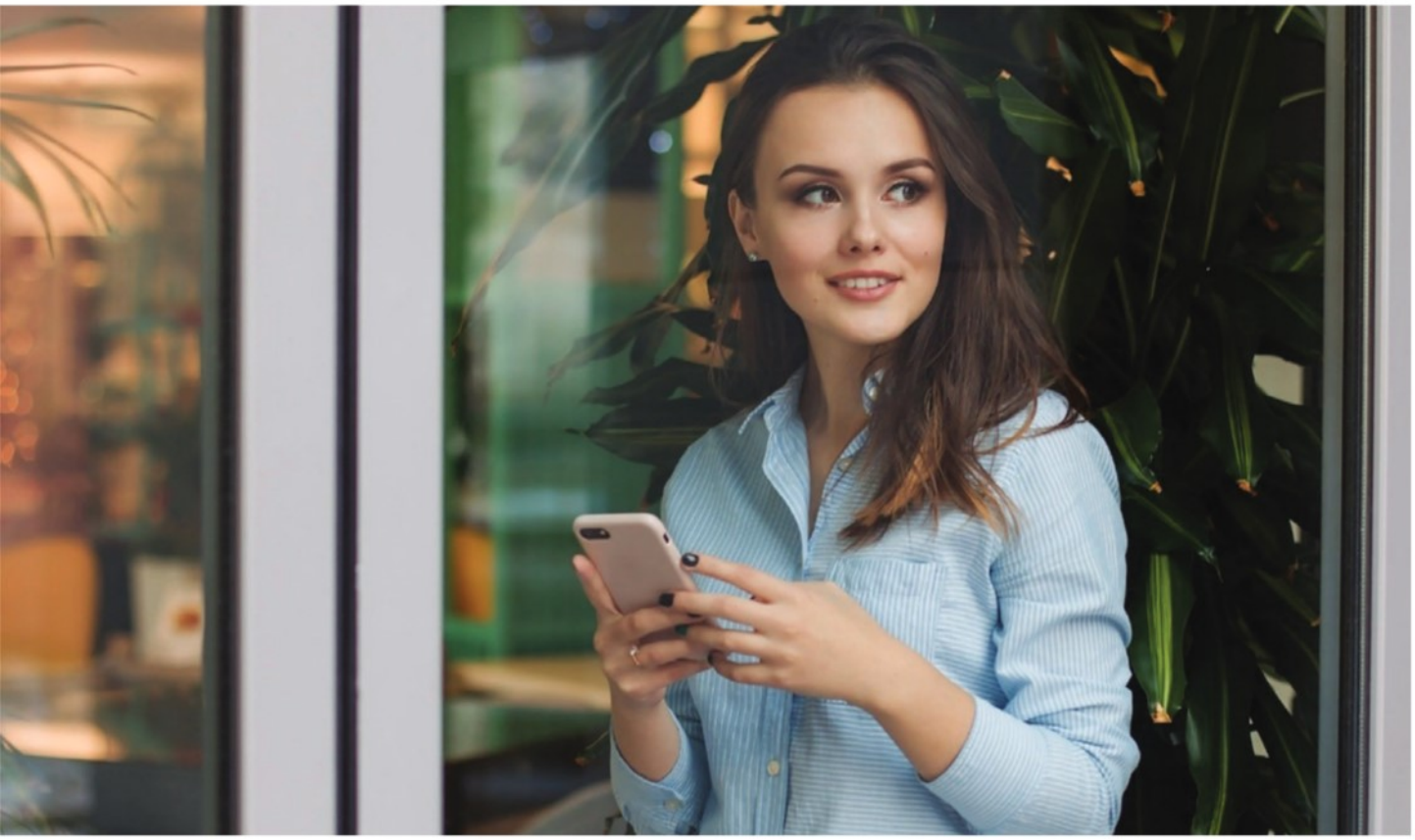
Often overlooked in the zealous application of "working software over comprehensive documentation" is the life-long work of Takeuchi and Nonaka captured in the Socialization, Externalization, Combination, Internalization (SECI) model.[10] First released in 1996, the model strongly endorses writing down tacit "knowledge" to make that knowledge "explicit" and therefore more easily shared and assimilated with other knowledge domains. The SECI model encourages the exact opposite behavior of the "documentation avoidance syndrome," a battle cry still of many Agile enthusiasts, whose fondness of product or process documentation is asymptotically null. Ironically, the earliest use of Scrum and rugby, and a non-waterfall-like approach for product development came 10 years earlier from the same pair of authors, Takeuchi and Nonaka in their seminal article *The New New Product Development Game*.[11] To highlight and chronologically sequence these events consider:

- 1986 – Takeuchi and Nonaka reference rugby and Scrum as inspiration for new work approaches.
- 1995 – Schwaber and Sutherland, influenced by the work of Takeuchi and Nonaka, present Scrum for software development at a conference in Austin, TX.
- 1996 – Takeuchi and Nonaka's publish the SECI model for capturing knowledge explicitly, that is, writing it down.
- 2001 – 17 software "leaders" convene and develop the Agile manifesto, which explicitly exalts working software over comprehensive documentation, seemingly contrarian with the SECI model authored by the same duo that also inspired Scrum as a framework. Schwaber and Sutherland were in attendance.
- 2010 – Schwaber and Sutherland (finally) release a written description of Scrum, 15 years after introducing it.
- 2020 – Schwaber and Sutherland release the current Scrum guide celebrating its shrinkage from 19 to 13 pages despite 25 years of Scrum learning.

To a lesser degree, function points suffer a similar fate. Specialization and compartmentalization have given rise to spin-offs of functional counting. Nonetheless, IFPUG's Function Size Measurement Method[12] captures the essence and steps for functional counting enveloped in a standard since 2009. Thus far, SNAP[13] has not encountered a splintering audience of stakeholders to the extent of competing standards. Competition is often healthy for communities of practice spawning fresh ideas. Sometimes less healthy is the re-spinning of the past with a twist, sometimes for fame or a few more followers.

Kanban has become a trendy, if not the fashionable *framework* for workflow management, per Kanban activists. But the use of Kanban has grown marginally by 2-3% since 2017 and its close cousin ScrumBan by the same slim growth since 2013.[14] Distancing ourselves from words like processes and methodologies, the use of the word *framework* is used to characterize concepts, values and principles, that suit less formally structured workgroups. Teams may substitute Scrum or task boards with Kanban boards, though they often dis-include work-in-progress (WIP) limits that help to ensure flow when optimizing cycle time. The fundamentals of flow, pull, WIP limits, and value stream optimization are seldom evidenced among Kanban teams. Teams are rarely self-organized and cross-functional—two of six "characteristics" Takeuchi and Nonaka recorded.[11] Kanban column heading become silos of specialties diminishing the impact of flow and further prolonging the paradigm shift to generalists. The lack of defined roles is often supplemented with titles that conflict with self-organization and team responsibility for product. The absence of collaboration (see Agile Principle 4) from undefined meetings leads to the adoption of scrum-like stand-ups, planning and grooming sessions, as well as reviews and retrospectives—the lack of which impairs the 10th Agile Principle. In all, eight of the 12 Agile principles that impact people and teams go unaddressed in Kanban, in part due to a lack of a standard (ISO, IEEE, ANSI, etc. as examples).

Scaling occurs naturally when teams need to share dependencies, risks and release components among each other. Yet scaling too has evolved into an array of customizations and hybrids. Certainly, there are market leaders with Agile scaling "frameworks." Most



are built around Scrum and Kanban, which should give one pause for concern given the maturity of most organizational practices. With more than 80% of organizations using one form of Scrum or Kanban, a mere 16% report a high level of Agile competency or practices enabling greater adaptability.[15] Scaling with unstable or less defined frameworks could be compared to setting a building's foundation on sandy soil. A few considerations when scaling include:

- Scope of effort: projects, programs, portfolios, organization(s), enterprise
- Fully committed business co-ownership; pigs only need apply
- The loss of agility at the team level as the organization scales "up" (e. g., sizing, iteration durations, releases and dependencies, team practices, tools, and feeding reporting hierarchies)
- Affordability given the overhead associated with new architect and engineer roles, unique values and principles sometimes portrayed as "lean"
- Assessing whether to scale to fit the culture or scale to upheave the culture
- The assortment and cost of new certifications, some with annual renewals
- Where to invest training dollars: becoming better at value delivery for your business or enhancing your expertise around scaling (hint: they may not be the same!)
- Using less sophisticated scaling approaches that align with Scrum-based organizations and that may be a more natural transition based on the needs of the organization

Turning to the last phrase "we use a tool" is cause for "alert." Over at least the past five decades, tools have been used to jumpstart organizational software engineering practices. In many cases tools have served as a substitute for the underlying knowledge necessary to develop and sustain more mature engineering practices. Leadership is often pacified if not enamored, with dashboards and colorful trending charts. Unforgettable was a presentation in 2006 where I first heard the saying "a fool with a tool is still a fool.[16]" That phrase can be traced to Ronald Weinstein not later than 1989[17]; it was used in the context of medical research.

Tools aren't the issue; not knowing what you want to achieve as an organization and believing that a tool will help define that—that's an issue! Tools are facilitators for intentional and purposeful processes, but their upkeep and feeding can distract from the value-added work of the organization and be a burden to teams. They often replace the needed daily face-to-face interaction (Agile Principle 6) with reminders from the Scrum Master or (hopefully not) a project manager to update tasks and hours in a tool so more accurate status charts can be viewed. Accountability among self-organized face-to-face team members gets supplanted with email reminders around "due dates and task times." Virtual meetings

become a hangout for disinterested parties to "multi-task" rather than participate. Collaboration is diminished. Cross-functional growth is thwarted. Competence plateaus. Teaming becomes another apparent victim of the pandemic aftermath. Increasingly, tools can replace the thinking and engagement that was the heart of our value delivery.

Standardization has enhanced almost every area of our lives: devices, appliances, housing, power sources, fuel, therapies, product quality, food, nutrition, automobiles, car seats, even software. ISO alone has more than 21,000 different standards developed by hundreds of working groups from almost every country.[18] The absence of documented process and products, the rudimentary application of sparsely-depicted frameworks, the stealthy displacement of team engagement by tools that become our process, the urgency to scale outcomes that disrupt team-level agility are all exacerbated without some standardization.

The benefits of standardization such as context, consistency, completeness and correctness are worthy of our attention.

By contrast, the adoption of IFPUG's Function Point Analysis for use in governments for software planning, estimation and costing has been recognized from Brazil to Italy, Japan, South Korea, Mexico, Malaysia and Poland. International standards have a global impact on defining activities and product realization. Pioneers like Takeuchi and Nonaka remind us to transform the tacit to the explicit, discovery to knowledge, experimentation to learning, data vagueness to information richness, unpredictability towards certainty, team formation into team performance and tolerance for mediocrity into market dominance. Please don't tell me you're doing...

REFERENCES:

[1]Keep the Baby (Examining Agile); Schofield; *MetricViews;* International Function Point Users Group; Winter, 2014 / 2015

[2]https://agilemanifesto.org/; retrieved 3/22/2021

[3]IEEE/ISO/IEC 12207-2017 - ISO/IEC/IEEE International Standard - Systems and software engineering -- Software life cycle processes

[4]ISO/IEC 9126:2001 Software engineering – Product quality

[5]http://vhg.com/standards-compliance/sw-cmm-software-engineering-capability-maturity-model/

[6]https://scrumguides.org/revisions.html; retrieved 3/22/2021

[7]https://agileforgrowth.com/scrum-history/

[8]https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/

[9]https://www.scrumstudy.com/sbokguide

[10]https://www.knowledge-management-tools.net/knowledge-conversion.php

[11]https://hbr.org/1986/01/the-new-new-product-development-game

[12]https://www.iso.org/standard/51717.html

[13]https://www.ifpug.org/about-snap/

[13]https://standards.ieee.org/standard/2430-2019.html

[14]Annual State of AgileTMReport; VERSIONONE, digitial.ai; 2013 – 2020

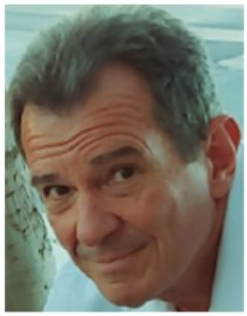
[15]Annual State of AgileTMReport; digitial.ai; 2020; pgs. 10, 9

[16]*A Fool with a Tool: Improving Software Cost and Schedule Estimation;* Ian Brown; ISMA Conference; San Diego, CA.; September, 2006

[17]*Perestroika, fashion, and the universal glue;* W M Thurlbeck; The American Review of Respiratory Diseases; 1989 May; quotes Ronald Weinstein, eminent academic pathologist in 1989 "a fool with a tool is still a fool"

[18]https://www.quora.com/How-many-ISO-Standards-are-there

ABOUT THE AUTHOR



Joe Schofield has more than 80 published books, papers, conference presentations and keynotes including contributions to the books The IFPUG Guide to IT and Software Measurement, IT Measurement, Certified Function Point Specialist Exam Guide, The Economics of Software Quality and his recently released Aligning People and Culture for Agile Transformation. He has facilitated ~200 teams in the areas of software specification, team building, organizational planning and Agile transformation. He holds eight Agile-related certifications: SA, SCT™, SSMC™, SSPOC™, SMC™, SDC™, SPOC™ and SAMC™. Joe was a CMMI Institute-certified instructor, an IFPUG Certified Function Point Specialist (CFPS) and a Lockheed Martin-certified Lean Six Sigma Black Belt. He is a past IFPUG President and for more than a decade he served as the "Chief Process Officer" of an organization of 400 software engineers.